

Buffer overflows, finishing pointer arithmetic, structs and linked lists

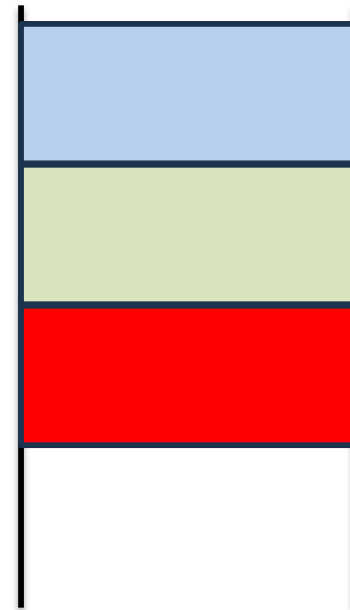
10/8/25

Administrivia

- HW 4 (histogram generator) due tonight

Buffer overflow lab

```
int fun() {  
    char s[5];  
    scanf("%s", s);  
    return strlen(s);  
}
```



Recall: Pointer arithmetic for strcpy

```
void f(char* a, char* b) {  
    while((*a++ = *b++));  
}
```

What about this?

```
int g(char* a, char* b) {  
    while(*a++ == *b++)  
        if(*(a-1) == 0)  
            return 0;  
    return *(a-1) - *(b-1);  
}
```

C structs

- Basically classes without methods

```
struct listNode {  
    int value;  
    struct listNode* next;  
};
```

C structs

- Basically classes without methods

```
struct listNode {  
    int value;  
    struct listNode* next;  
};
```

- Must have semicolon at the end (or it thinks you're declaring one)
- We'll typically use typedef to create an alias:

```
typedef struct listNode node;
```

- Access fields with a dot:

```
printf("%d\n", n.value);           //assumes n is a node
```

Which of the following lines will the compiler flag as a syntax error?

struct obj { //A

int x; double y; //B

}; //C

...

obj s; //D

s.x = 10; s.y = 2.3; //E

Which of the following lines will the compiler flag as a syntax error?

```
struct obj {  
    int x;  
    char s[10];           //A  
}
```

```
int val = 3;              //B
```

```
int main() {              //C
```

```
    struct obj st;  
    if(st.x)               //D  
        st.s[0] = 0;      //E
```

```
}
```

Write some linked list code

and learn to love arrows ->